This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1- 14. (Canceled)

15. (Currently amended) A method for streamlining operations in a coprocessor by indicating whether all required memory resources for a task are available ~~prior to beginning the processing of the task~~, comprising:

prior to beginning the processing of the task,

preparing a task for processing in a coprocessor by paging memory resources, within a memory space accessible by the coprocessor, associated with the task into coprocessor-readable memory;

sampling the memory resources to determine if all required memory resources are in a proper location in the coprocessor-readable memory;

recording whether all required memory resources are in a proper location in the coprocessor-readable memory, wherein said recording generates an indicator memory resource that is associated with the task; and

processing said indicator memory resource substantially at the beginning of processing the task, wherein if said indicator resource indicates that all required memory resources are not in ~~a~~ the proper location in the coprocessor-readable memory, the coprocessor stops the initiation of the processing of the task.

16. (Currently amended) A method according to claim 15 wherein the coprocessor is a Graphics Processing Unit (GPU).

17. (Original) A method according to claim 15 wherein the task is represented by a DMA buffer.

18. (Original) A method according to claim 15 wherein the coprocessor stops processing the task because processing said indicator memory resource generated a page fault.

19. (Original) A method according to claim 15, further comprising maintaining a list of tasks that the coprocessor stopped processing, so that all required memory resources can be brought to a proper location in coprocessor-readable memory at a later time.

20. (Original) A method according to claim 19 wherein the later time is determined based on a priority of tasks on the list of tasks.

21. (Original) A method according to claim 20, further comprising a periodic priority boost that increases the priority of one or more tasks on the list of tasks to ensure that all tasks eventually can be processed.

22. (Original) A method according to claim 15, further comprising generating a page fault when a context switch occurs to a context that references an invalid ring buffer or an invalid DMA buffer.

23-37. (Canceled)

38. (New) A method for streamlining operations in a coprocessor by indicating whether all required memory resources for a task are available, comprising:

prior to beginning the processing of the task,

preparing the task for processing in a Graphics Processing Unit (GPU) by paging memory resources within a memory space accessible by the GPU associated with the task into GPU readable memory;

sampling the memory resources to determine if all required memory resources are in a proper location in the GPU readable memory;

recording whether all required memory resources are in a proper location in the GPU readable memory, wherein said recording generates an indicator memory resource that is associated with the task; and

processing said indicator memory resource prior to beginning processing the task, wherein if said indicator resource indicates that all required memory resources are not in the proper location in the GPU readable memory, a page fault is generated, the task is not processed and a list comprising tasks that the coprocessor stopped processing is maintained.

39. (New) A method according to claim 38 wherein the task is represented by a DMA buffer.

40. (New) A method according to claim 38 wherein the coprocessor stops processing the task because processing said indicator memory resource generated a page fault.

41. (New) A method according to claim 38 wherein the later time is determined based on a priority of tasks on the list of tasks.

42. (New) A method according to claim 41, further comprising a periodic priority boost that increases the priority of one or more tasks on the list of tasks to ensure that all tasks eventually can be processed.

43. (New) A method according to claim 38, wherein said page fault is generated when a context switch occurs to a context that references an invalid ring buffer or an invalid DMA buffer.